Approved for public web release

DII COE

4.x Kernel Overview

Presenters: Dr. Thom McVi
Chief Archite
Randy Heath
Lead Develo
Mike Pajevski
Security Eng
Calvin Miyazon
Task Manage

DII COE Developers' Technical Interchange
23, 24 May 2000

AGENDA

- Overview
  - Drivers for Change
  - Key Changes
  - Deprecated 3.x Kernel APIs
- Detailed Discussions
  - COE Installer and I&RTS changes
  - Segment Dependencies
  - Environmental Inheritance
  - Process Management
  - Setting up Domains in APM

Drivers for Change

4.x must provide support for Additional Platforms

- NT 4.0
- 18 KCP platforms

Review our kernel approach during major revisions & align with best industry practice.

- Goal: Move from GOTS to COTS
- Identify areas where we differ from industry.
- Evaluate whether the differences justify GOTS components.
- Upgrade our technology and approach

Integrator flexibility

- provide integrators with greater flexibility in how they combine and field segments.

Address Security Issues found in 3.x

Guiding Rules

- Must provide Unix and NT parity.
- Must allow 3.x segments to execute correctly on a 4.x platform.
- Must provide a "way ahead" to a "mostly-COTS" kernel.

Key Changes in 4.x

Removal of Account Groups

- Introduce the concepts of "segment dependencies" & "services".
- Changes in how a process' environment is established & how processes are launched.
- Changes in how profiles are defined.

Inclusion of a Common Data Store (CDS)

- Common repository of most kernel related data (replaces flat files).
- Can also be used by COE and Mission Applications.

Account & Profile Manager (APM)

- Support for NIS+ and NT domains & local accounts on Solaris, HP, & NT.
- Works with native O/S calls.

Key Changes in 4.x (cont.)

COE Installer

- Consistency with I&RTS
- Supports additional "integrator" directives (bind, process groups).
- Security Modifications - changes in how permissions and ownership for files are determined.

Process Management

- Removal of Process Executive
- Changes in how boot, session, background and transient processes are launched.
- Changes in how the environment for each of these process types is established.

Key Changes in 4.x (cont.)

Features

- Provides abstract kernel-level support for menus, icons, and permits
- Allows additional interface metaphors to be added by integrators/sites.
- Will be abstracted further in 5.x

Security Enhancements

- Support for a choice of user shells
- Permission lockdowns on various o/s native services and files

Kernel Application Architecture

- Most components are now 3-tier
- Java GUIs and business rules
- Abstract services provided by a set of native APIs

Deprecated 3.x Kernel APIs

- **Identified in the I&RTS as deprecated.**
- **Supported in 4.x, but NOT in 5.x**
- **Currently undergoing AOG approval**
- **Summary areas:**
  - **User Profile API's**
  - **Special Utilities for Compatibility API's**

Where are we going in 5.x?

COTS-based Software Installer

- Java based extensible installer framework (JSR-38)
- Small, or zero footprint.
- ZeroG, open source Unix developers, Sun, IBM
- Will allow DISA/services to purchase DII COE compliant installers from a variety of different vendors.

Support for additional platforms

- Real-time (beta 1 already in release)
- Larger machines (HP, Solaris)

Configurable Kernel

- Kernel delivered as a set of bundled segments.
- Can install all or only part that mission applications require.

Where are we going in 5.x ? (cont.)

Exposed set of "kernel" APIs

- Native code also callable via .jni and likely CORBA (IDL)
- Good initial adoption by industry

Ability to consistently update the operating system version independently of the kernel version.

Detailed Discussion Areas

- **COE Installer and I&RTS changes**
- **Segment Dependencies**
- **Environmental Inheritance**
- **Process Management**
- **Setting up Domains in APM**

Installer/I&RTS Modifications

- Deprecated descriptors.
  - [AcctGroups]
  - [Permissions]
  - [ReqrdScripts]
  - [SegType]
  - [SharedFiles]
- Additional SegDescrip files (VerifySeg warns if absent).
  - FileAttribs (plus $SegDir).
  - Integ/IntgNotes.
- Comp_Table is ignored.
- FilesList is optional for NT COTS segments.

Segment Permission (UNIX)

- Segment is installed with the original ownerships and permissions
  - The values set on the file when MakeInstall is run.
- All unknown owner UIDs are set to COE (uid 400).
- Unknown group GIDs are not modified.
- "World writeable settings" are removed (o-w).
- FileAttribs settings are applied.
- Always use FileAttribs!
  - If you don't, installed permissions may vary depending on the settings when MakeInstall was run.

Segment Permissions (NT)

- Inherits the ownership and permissions of the installation directory or file system (if installed at root).
- The DII COE sets "h" to be owned by the group "Administrators".
- The DII COE sets "h" to have these permissions:
  - Administrators have Full Control
  - System has Full Control
  - Authenticated Users have Read (RX)

Writing "Good" 4.x Segments

- "Good" ≡ minimizes problems during eventual transition to DII COE 5.x.
- Do not use any deprecated descriptors.
- Do not use $BACKGROUND, $SESSION and $SESSION_EXIT processes.
  - Transient processes should check for the existence of required support processes and start them as required.
- Do not rely on DII COE to set process environment.
  - Process should establish their own environment.
  - Processes should not use environment variables to communicate information to other processes. Use a registry-like service (e.g. CDS) instead.
- Provides Integ/IntgNotes (NT and Unix) and FileAttribs (Unix).

Process Environment

Process	UNIX	NT
$BACKGROUND	User environment Script file	None
$BOOT	Bourne shell minimums Script file	None
$PERIODIC	Bourne shell minimums Script file	None
$RUN_ONCE	Bourne shell minimums Script file	None
$SESSION	User environment Segment closure [ReqrdScripts]	None
$SESSION (Process Group)	User environment Segment closure [ReqrdScripts] Script file	None
$SESSION_EXIT	User environment Remaining segment closure [ReqrdScripts]	None
Transient	User environment Segment closure [ReqrdScripts]*	None

* If launched using /h/COE/bin/COE_launch
Bourne shell sets HOME, LOGNAME, PATH, PWD, TZ, USER

User Environment includes $HOME/.login, $HOME/.cshrc, /h/COE/Scripts/.login.COE, /h/COE/Scripts/.cshrc.COE

Definitions

Segment Dependencies

The list of segments that a segment directly calls during run-time.

This will be defined by a 4.0 required directive, or computed based on the structure of an existing 3.x segment and/or account group.

Segment Closure

The set of segments that provide at least one feature that is in the user's active profile set.

Segment Dependencies & Segment Closure

- In 3.x, Account Groups provided the basic structure for:
  - establishing a process' environment.
  - launching any required session processes.
- In 4.x, we use Segment Dependencies & Segment Closure to meet these needs.
  - Transitional step towards use of industry standard process launch mechanisms in 5.0 and beyond.
- For 3.x segments:
  - The 3.x to 4.x Kernel Upgrade and 4.0 COE Installer will properly handle the segment. No action is required.
- For 4.x segments
  - New descriptor defined in I&RTS 4.0
  - Requires segments to state explicit run-time dependencies (aka services)

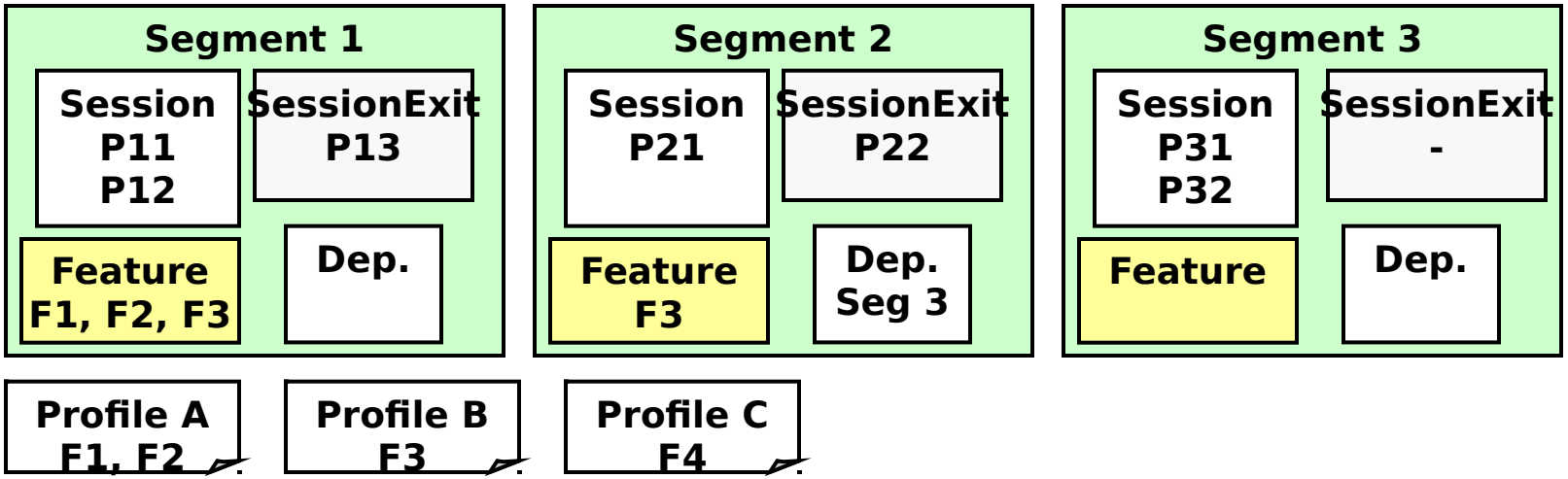
How it Works

- Segments deliver services, processes and features.
- Features are assigned to profiles.
- Users assume profiles. These profiles are the user's active profile set.
- We compute Session Closure by examining the user's active profile set, and generating a list of segments that are represented by at least one active feature.
- We then use the dependencies specified by those segments to compute a complete segment dependency graph. The kernel will then traverse this graph to establish the necessary process environment & launch the necessary session processes.

Session Process Environment

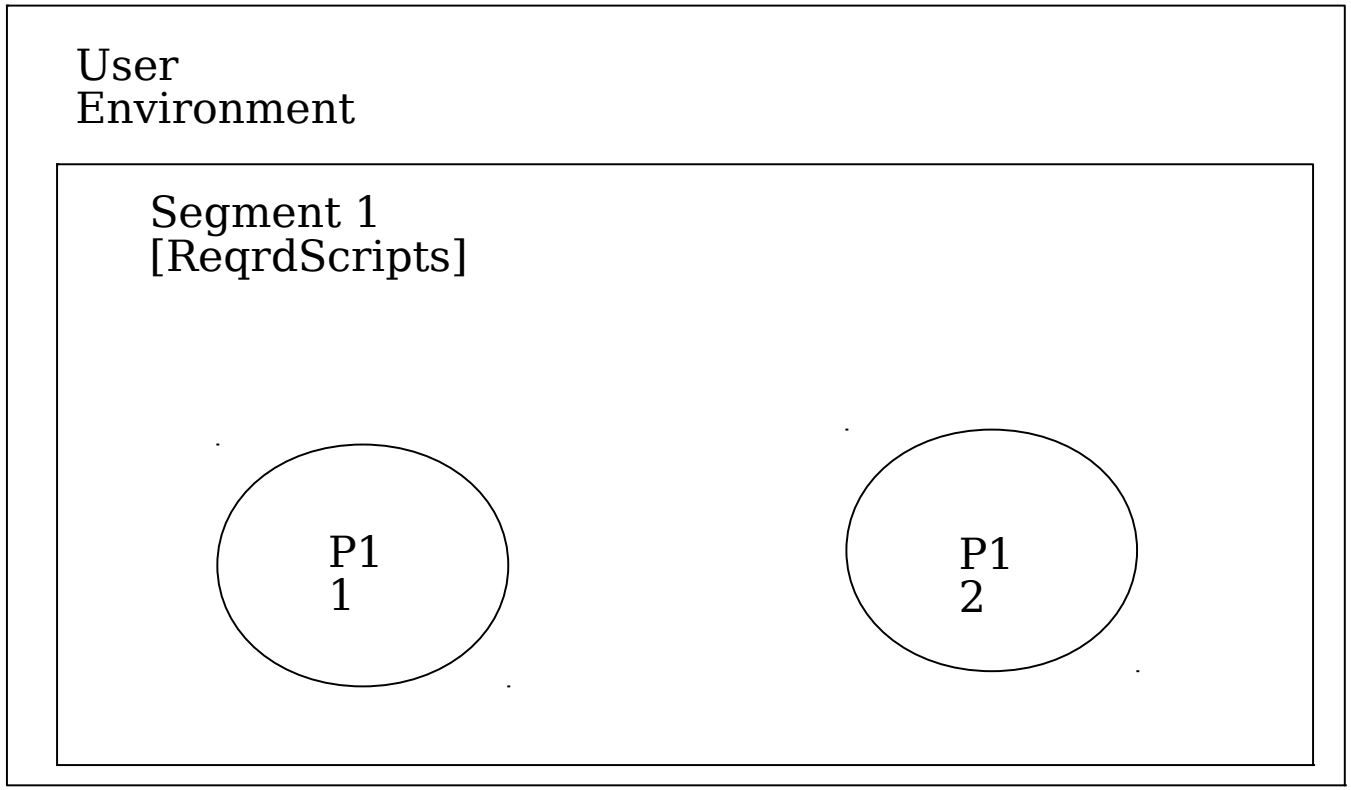
- Segment Closure is computed when the user submits his profile selections through the profile selector.
- All session and session exit processes started by a particular use of the profile selector will get the same segment environment.
- Session processes established by [ProcessGroup] may also get additional environment values.
- Session processes will get their segment's and dependent parent's segment environment.
- Session processes will also get the environment of unrelated (peer) segments.
- In the examples, the order of environment variables goes from outermost to innermost.
- Peer environment order is non-deterministic. (Noted with "ND")
- Note: The user's profile choices can change the environment for new session processes and transient processes.

Session Closure Example

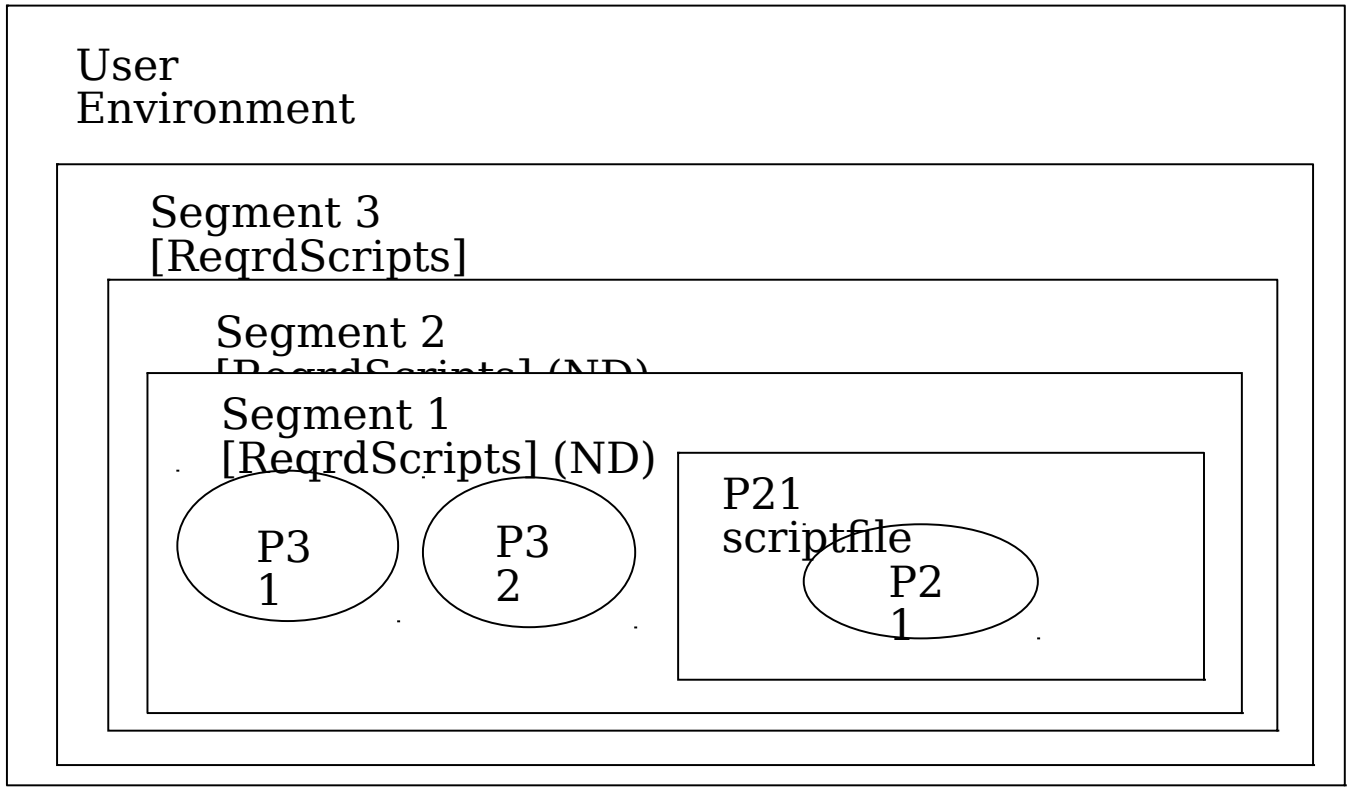


- User Assumes Profile A
  - segment closure contains {segment 1}, no other segment dependencies
  - launches P11 & P12
- User Assumes Profile B
  - segment closure contains {segment 1, segment 2}, adds dependency on segment 3
  - launches P31, P32, P21
- User Drops Profile A
  - segment closure contains {segment 2}, adds dependency on segment 3
  - executes P13, kills P11 & P12

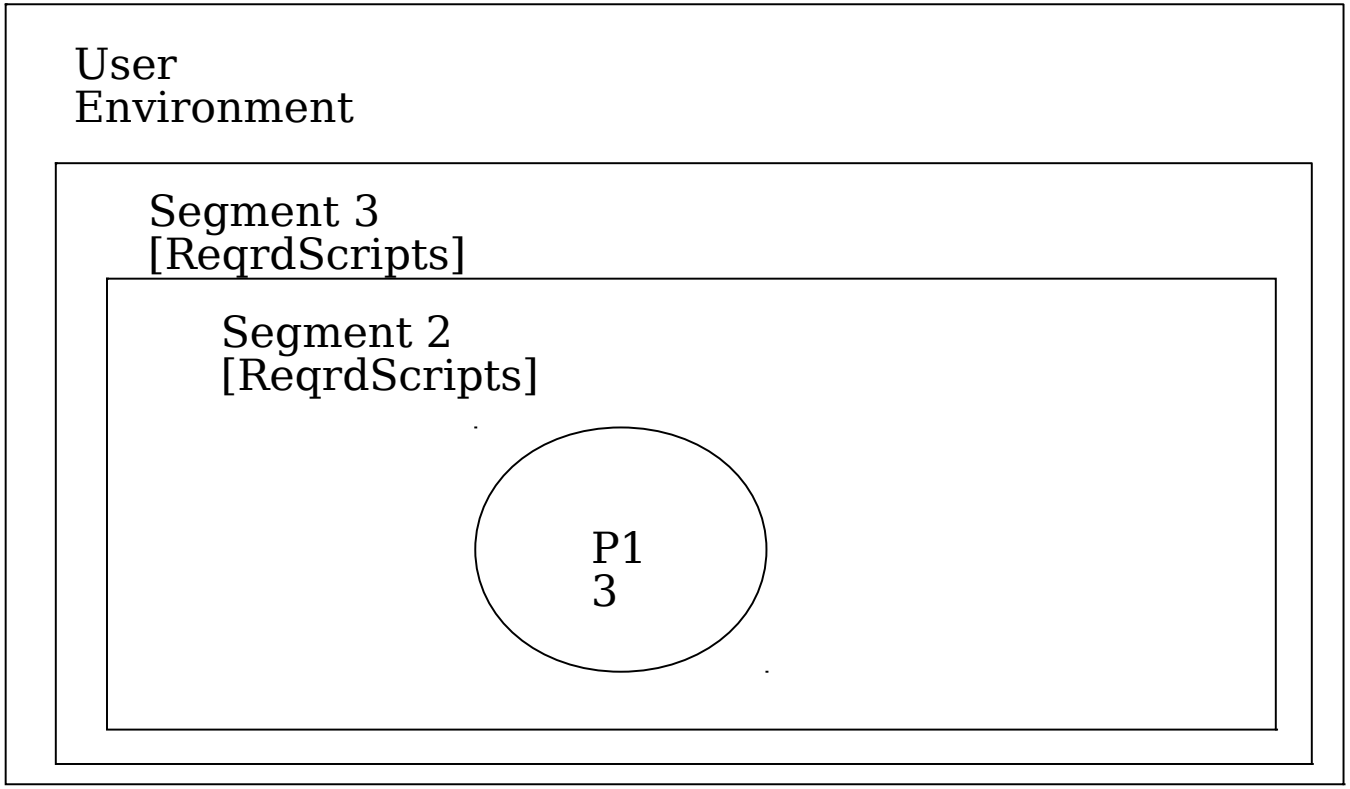
P11/P12 Environment Example



P31/P32/P21 Environment Example



P13 Environment Example



Debugging Segment Processes on Unix

- Verify that the environment is being set as you expect.
  - Create a test script that directs the environment to a file.
  - "env | sort > /tmp/somefile; id >> /tmp/somefile"
- Verify that the process can run alone.
  - create a test script that sets the expected environment and runs the application.

Debugging Segment Processes on Unix (con)

- $BACKGROUND
  - Remove /var/tmp/.APM_bg_processes_were_run.
  - /h/COE/Comp/APM/bin/APM_run_firstlogin
- $BOOT
  - /etc/rc3.d/S13coeinit start
  - /h/COE/Comp/Util/bin/COE_start_boot_processes
    - -v will show you the boot processes that it is running
    - -n will show you the boot processes that will be run, but will not actually run them

Debugging Segment Processes on Unix (cont)

- $RUN_ONCE
  - Check CDS /LocalHost/DII Kernel/Process/-<process name>
    - hasRun (if "false" means will run on next boot)
- $PERIODIC
  - check root's crontab
- $SESSION/$SESSION_EXIT
  - After assuming profile(s), check $HOME/../data
    - segList.:0 has list of segment [ReqrdScripts] that are sourced to establish the session environment.
    - cache.:0 has environment used for transient processes launched by COE_launch.
    - session.:0 has combined set of commands that establish the user's environment and launch session processes.

Debugging Segment Processes on NT

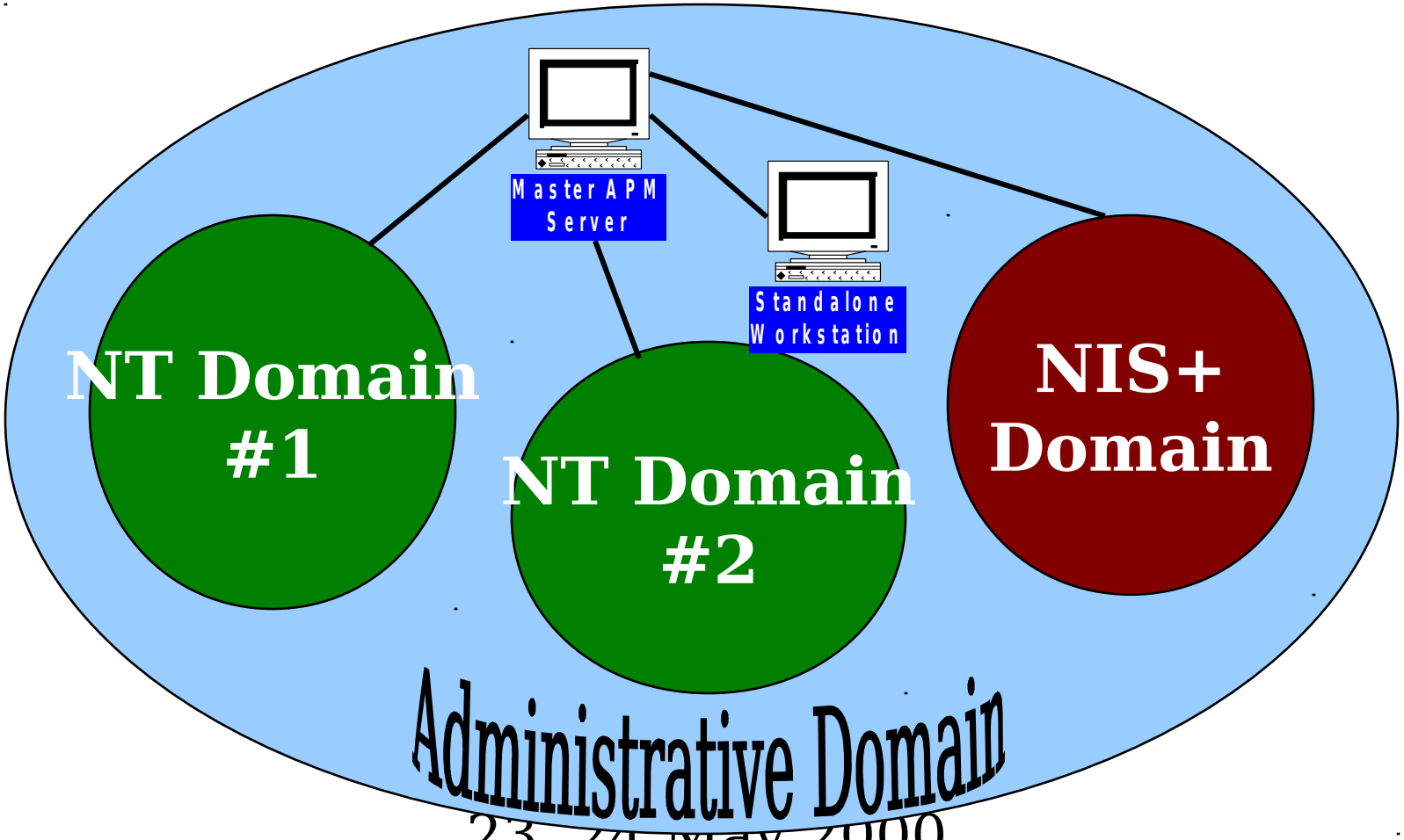
- %SystemDrive%\temp\session.log
  - Logs $BACKGROUND, $BOOT and $RUN_ONCE processes that were run.
  - Logs $PERIODIC processes that were scheduled.
  - Can verify $PERIODIC with "at" command.
- $SESSION/$SESSION_EXIT
  - Replace your program with a trivial program (e.g. COEMsg) to make sure that your program is being launched.
  - No environment variables are set by the Kernel.
  - Works best as an executable, not a batch command.

Administrative Domain Overview

- APM provides account management capabilities across the three DII COE platforms
  - HP-UX
    - Standalone workstations and servers
  - Solaris
    - Standalone machines and NIS+ domains
  - Windows NT
    - Standalone machines and NT Domains
- Centralized management of an "Administrative Domain" via the APM Client GUI interface

The Administrative Domain

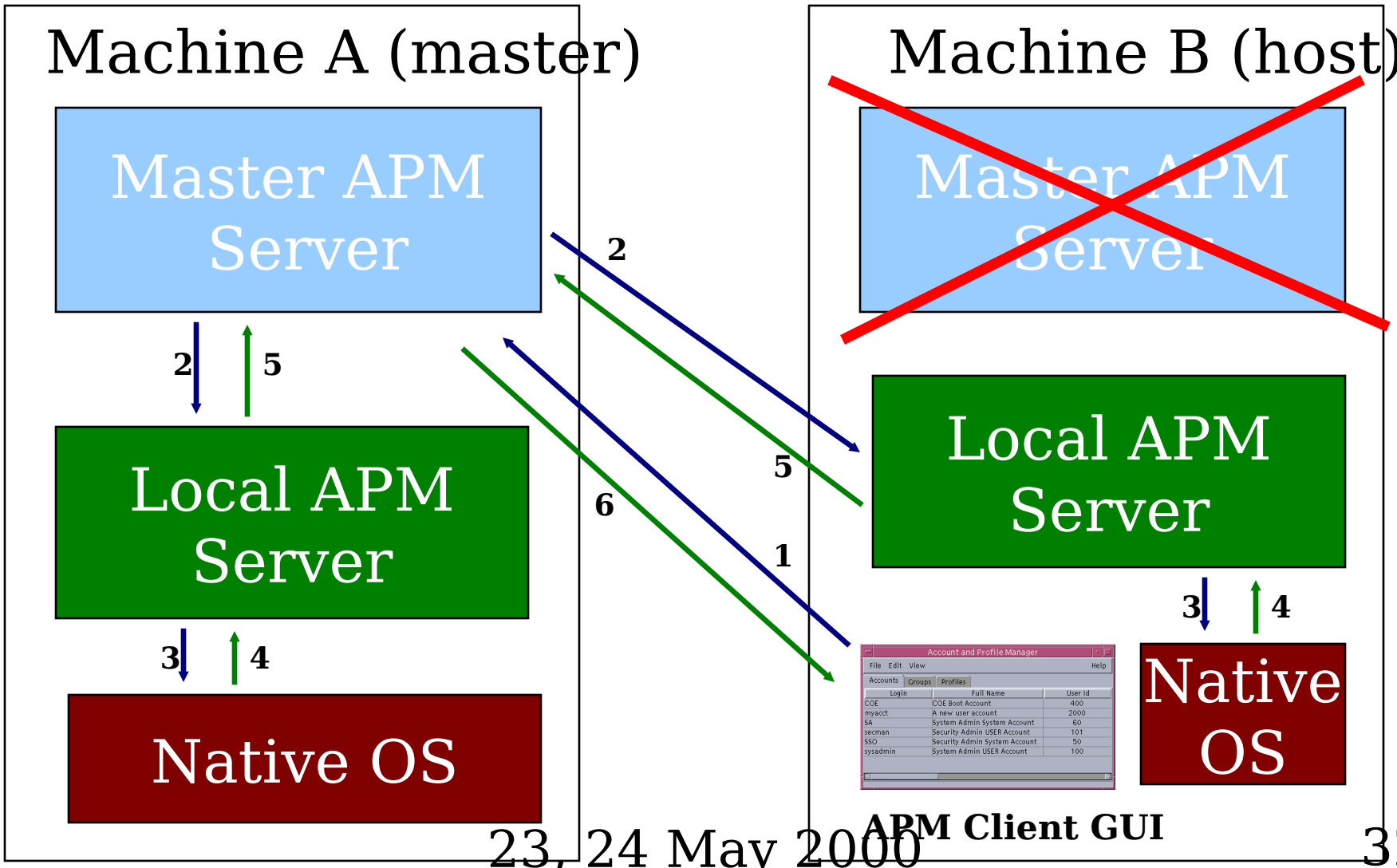
- Below is an example of an administrative do



APM Architectural Overview

- APM is a three-tiered architecture
  - APM Client GUI
    - Provides an interface to the Master APM Server
  - Master APM Server
    - Maintains records of all hosts, accounts, groups, profiles, and segments in the APM domain
    - Distributes commands (eg, to add/delete accounts) to the various Local APM Servers in the APM domain
  - Local APM Server
    - Processes commands received from the Master APM Server
      - Interfaces with the native OS to manage accounts and groups.
      - Interfaces with the local CDS to manage profiles and segments.

APM Communications Flow



APM Client GUI

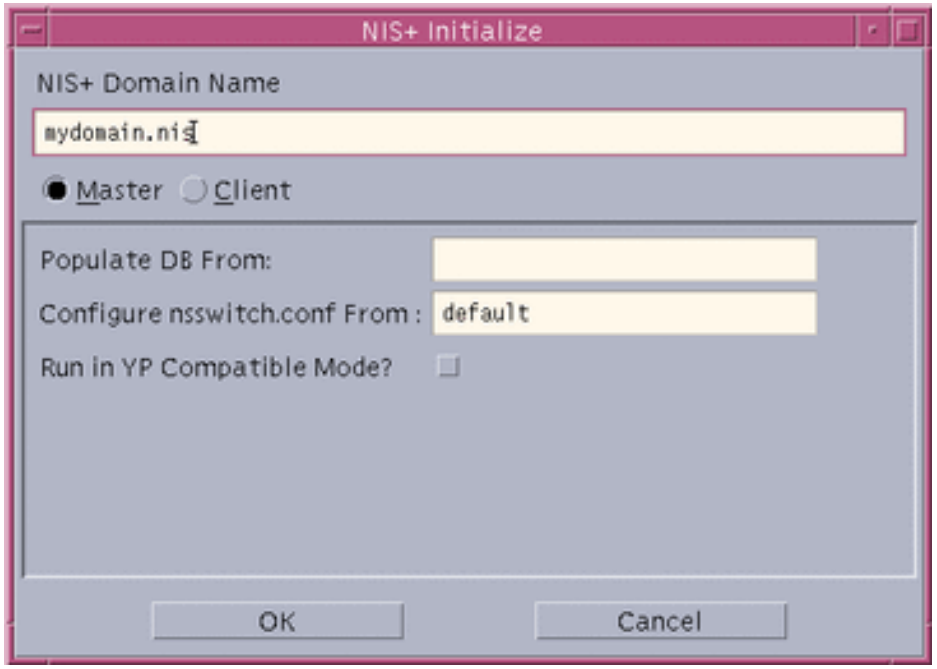
Building Administrative Domains

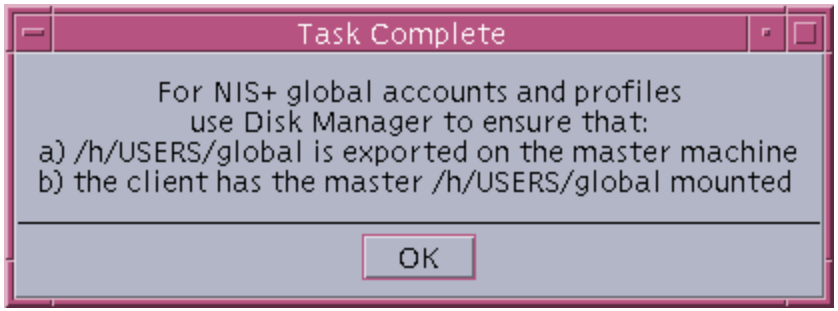
- DII COE machines start out as APM standalones
  - Use their own Master APM Server
    - Alone in their own administrative domain
  - May be part of an NT or NIS+ domain
- DII COE machines are "merged" to build an administrative domain
  - Tell new host which Master APM Server to use
  - Setup authentication keys on new host and master
  - Transfer information in the common data store (CDS)

Administrative Domain Setup Tools

- Kernel tools used to build administrative domains:
  - NIS+ Admin Tool
  - apm_register_pdc.exe
  - apm_register_pdc_client.exe
  - Edit APM Configuration
    - Identify which Master APM Server a host should use
  - Authentication Manager
    - New GUI provided in 4.2.0.0 P2
  - Merge Host

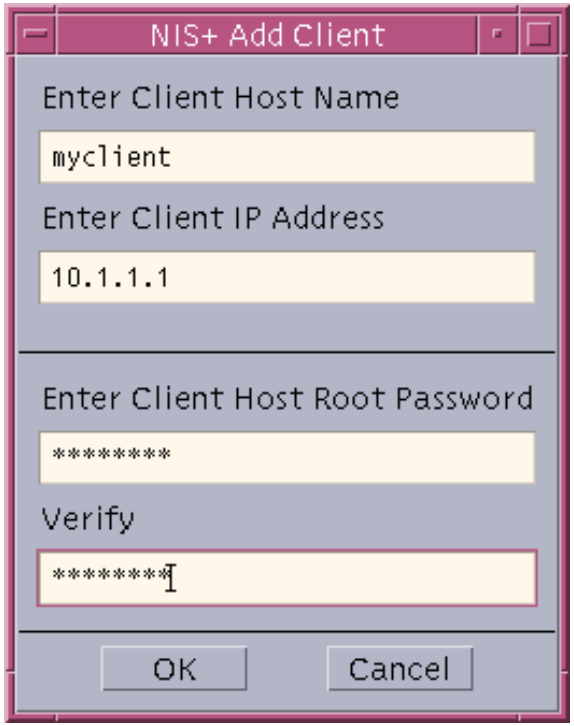
Initializing NIS+ Servers





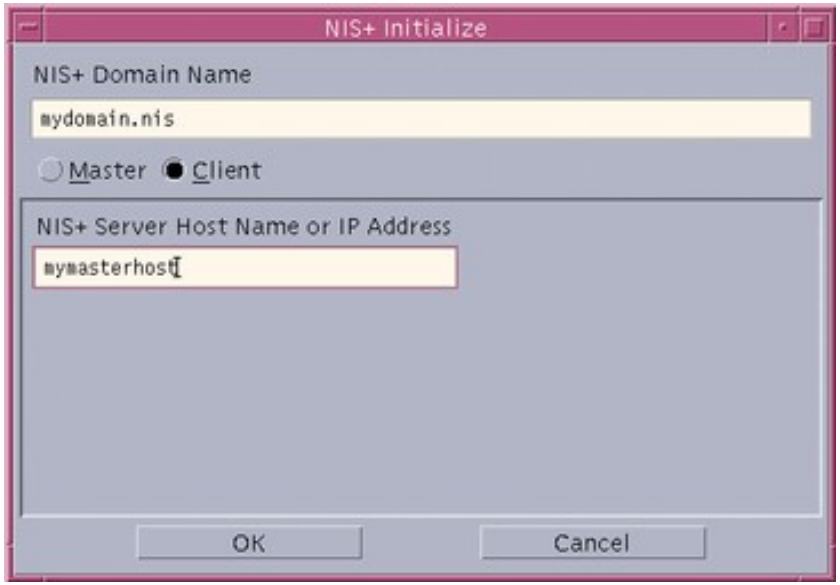
- "Initialize NIS+" GUI
  - Initialize as "Master"
  - Specify domain name
  - Populate from existing database
  - May run in yp mode
    - Security risk!
  - Message reminds administrator to export the global user directory

Adding NIS+ Clients



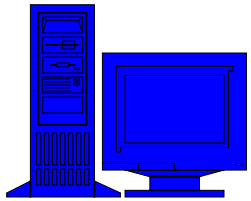
- "Add NIS+ Client" GUI
  - Run on NIS+ server
  - Informs NIS+ server of clients

Initializing NIS+ Clients



- "Initialize NIS+" GUI
  - Initialize as "Client"
  - Specify name of NIS+ master
  - Message reminds administrators to mount the global user directory that is located on the NIS+ master server

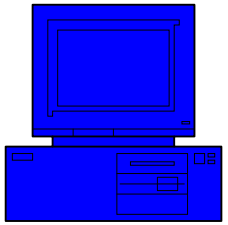
Configuring NT Domains



Primary Domain Controller

- Install the NT OS
- Install NT Service Pack
- Load the COE kernel
- Run:

  apm_register_pdc.exe



NT Workstation

- Install the NT OS
  - As member of the NT domain
- Install NT Service Pack
- Load COE kernel
  - As Domain Administrator
- Run:

  apm_register_pdc_client.exe

APM Authentication Overview

- APM uses a key-based authentication mechanism*
  - Each Local APM Server has a local authentication key
    - Used to authenticate the master APM server to the local APM server whenever transactions are performed (ie, adding accounts)
    - Randomly selected during kernel installation if APM authentication is enabled
    - Must be changed to a known value before merging machines
  - The Master APM Server has a master authentication key
    - Used to authenticate the user running the APM Client to the Master APM Server
    - Input by user during kernel installation if APM authentication is enabled

***Temporary mechanism in place until the Security Services Architectural Framework (S**

Setting up APM Authentication

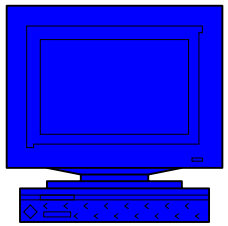
- The Authentication Manager is used to setup keys
  - GUI provided in 4.2.0.0 P2
    - /h/COE/Comp/APM/bin/APM_AuthMgr
  - Command line interface still exists
    - /h/COE/Comp/APM/bin/APM_AuthMgr <arguments>
      - Use -h to get a list of valid arguments
  - Must initialize APM Authentication if it was not enabled during kernel installation
  - On the new host (the machine that will be added to the administrative domain), set local key to a known value
  - On Master APM Server, add the new host and its key

Running Merge Host

- The Merge Host tool is used to transfer CDS information to the Master APM Server
  - Account, group, profile, segment, and domain (NIS+/NT) information is copied to CDS on the Master APM Server
  - Conflict resolution takes place during the merge process
    - Accounts, groups, and profiles with the same name are resolved
      - Options are: Use New/Master, Delete on New/Master, Customize
    - Conflict resolution choices have consequences
      - Example: Using the master's version of the SSO Profile when merging Solaris into an NT master will cause some of secman's icons to be lost on the Solaris machine
        - Should "Use New" instead to preserve characteristics

Merge Host Example: Step 1

Step 1: Point host to its new Master APM S

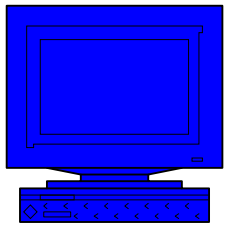


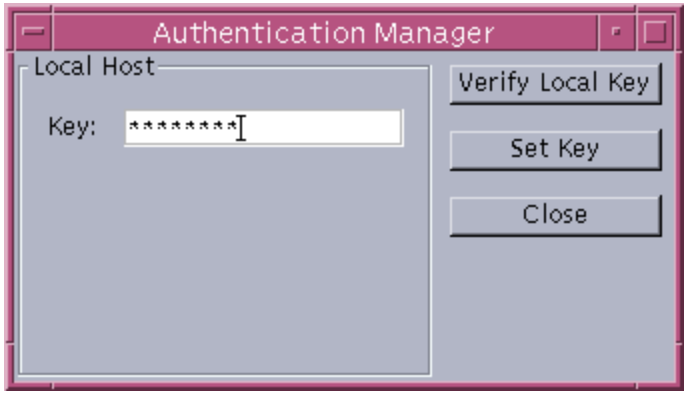
Tool: **Edit APM Configura**

Action: **Change the entry in the Master APM Server field to the name of the new master server.**

Merge Host Example: Step 2

Step 2: Setup authentication on the merging host.



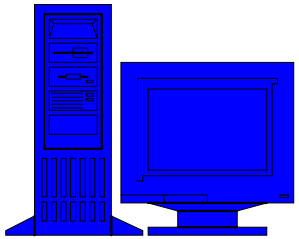


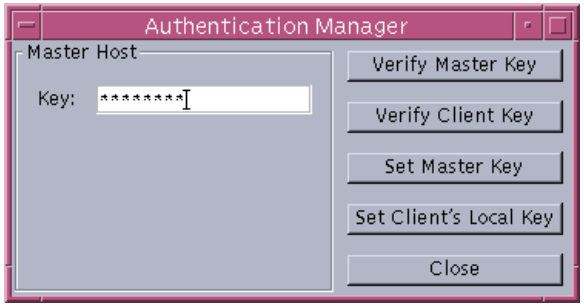
Tool: **Authentication Manager GUI**

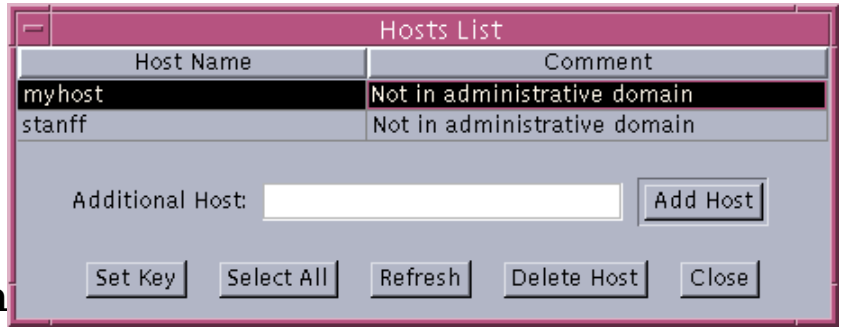
Action: **Change local authentication k**
to a known value.

Merge Host Example: Step 3

Step 3: Setup authentication on Master APM Server.





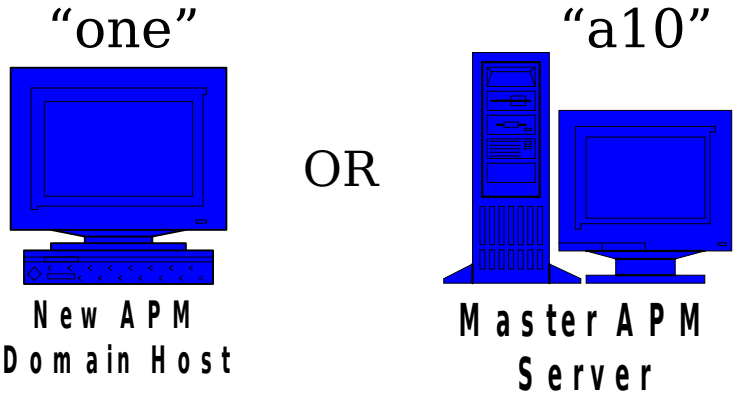


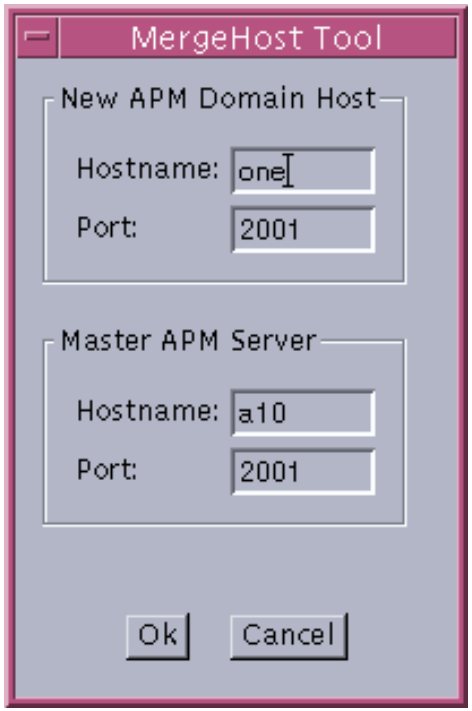
Tool: **Authentication Ma**

Action: **Inform Master APM of the new host's local authentication key.**

Merge Host Example: Step 4

Step 4: Run Merge Host.





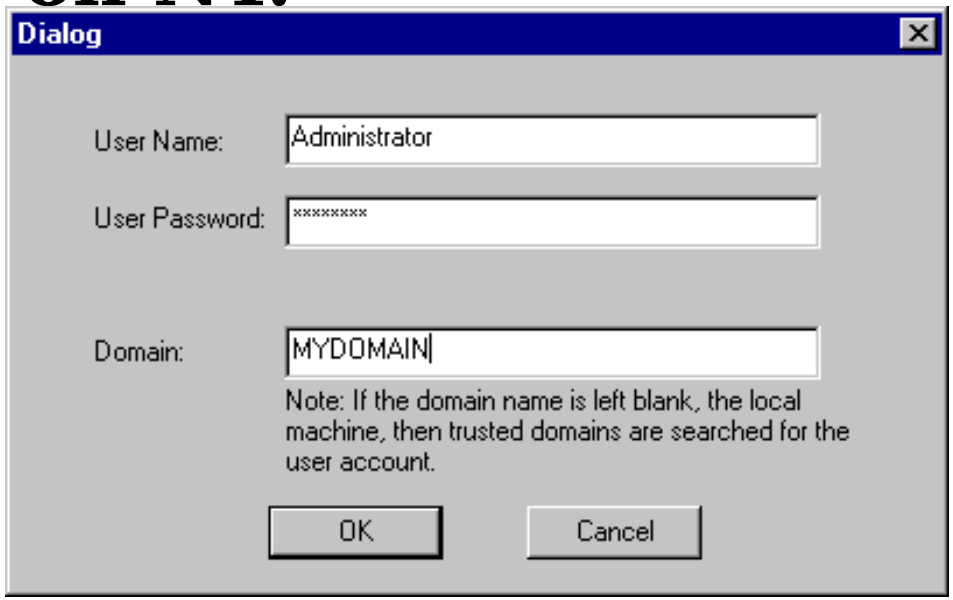
Tool: **Merge Host GUI**

Actions: **Fill in the fields (hostnames).**
Start the merge host process.

N OTE: All fields will be filled in automatically when this tool is ru on the New APM Domain Host.

Merge Host Example: Step 4a

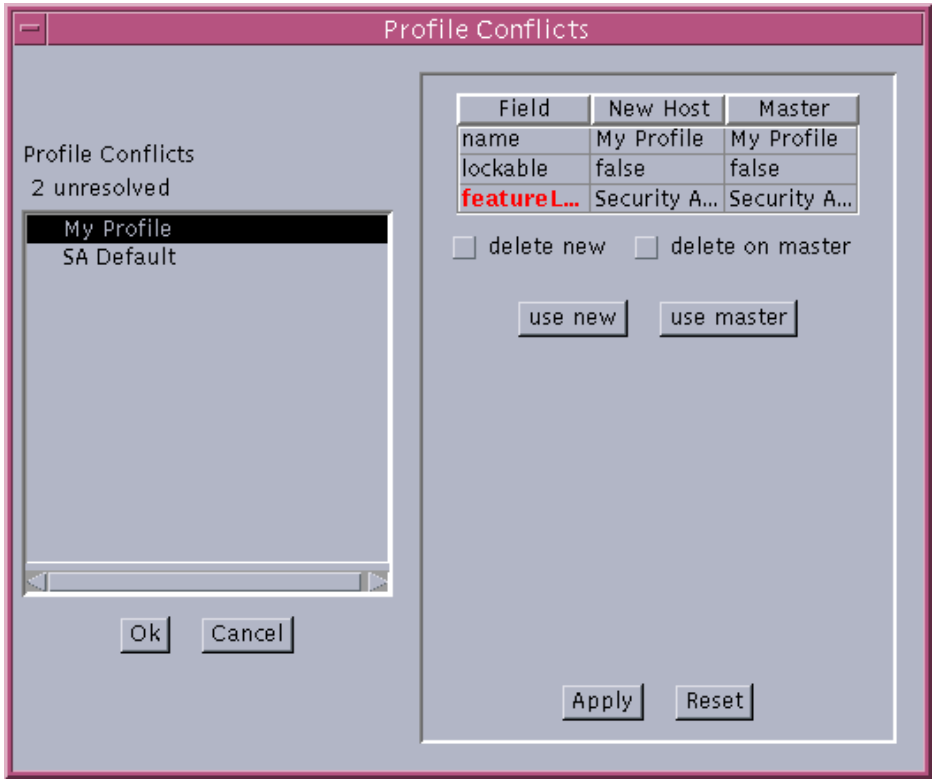
Step 4a: Provide Administrator password on NT.



NOTE: On NT only, when running Merge Host as secman (or any user othe Administrator), this dialog box will appear prior to the GUI shown on the p

Merge Host Example: Step 5

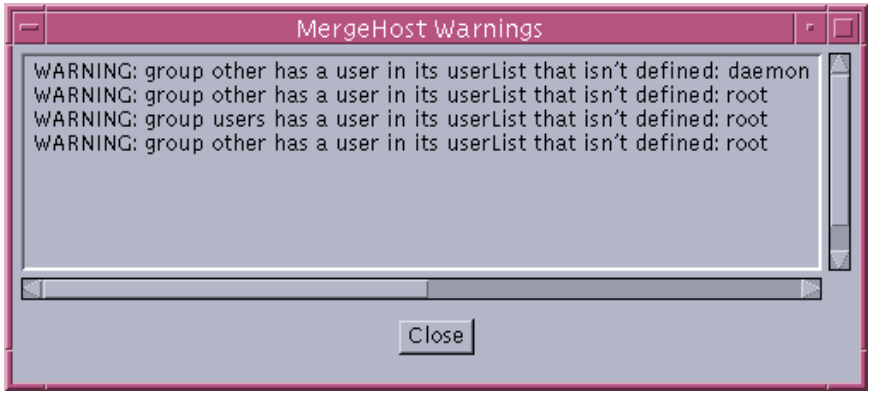
Step 5: Resolve conflicts.

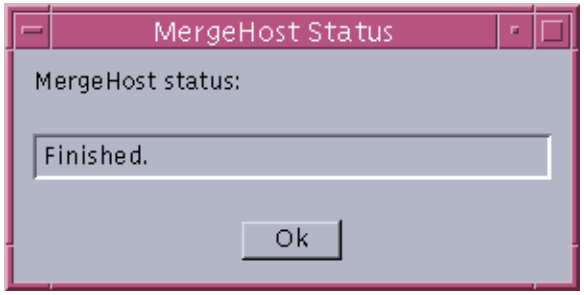


- Conflicts resolved for duplicate names of:
  - Accounts
  - Groups
  - Profiles
- Choices will have an impact on the entire administrative domain.

Merge Host Example: Step 6

Step 6: Finish Merge Host.





- Warnings are generated for minor issues that do not effect the process.
- SEVERE errors mean some data was not transferred.
  - Generally due to a default group not present on master.
  - Remedy is to repeat the process immediately

Merge Host Example: Step 7

Step 7: Resolving severe errors.

- Running Merge Host a second (or third…) time will result in the warning shown to the left.
  - Master APM Server already has knowledge of the client.
  - Click Continue to proceed.
- Second pass though should resolve all severe errors and complete the process.

Known Issues with APM Domains

- Many issues related to building APM domains have been resolved by the 4.2.0.0 P2 version of the kernel
  - NT/NIS+ domain information is properly transferred
  - NIS+ domains can be constructed before or after merging the individual machines into an administrative domain
- Some considerations remain
  - NT domains should be configured prior to merging
    - NT workstations should be made members of an NT domain prior to loading the kernel (unless they will always be in a workgroup)
      - Problems will arise if NT workstations are changed from workgroup to domain members after loading the kernel
  - Cannot create duplicate local and domain account names

Additional Considerations

- Some guidelines to follow
  - Merge domain controllers (NT/NIS+) before their clients
    - Tells APM about domains before telling it about domain members
    - Permits domain account management sooner
  - Wherever possible, load the kernel and configure the domain before creating accounts and loading segments
    - Simplifies administration
      - Less conflicts during merges
      - Define data sharing (exporting and mounting shares) strategy early
      - Setup and verify APM authentication up front (rather than enabling it at some later date)

Summary and Conclusions

- APM provides centralized management of administrative domains
  - Can contain various platforms and their domains
- Three-tiered architecture
  - Each machine has a client (GUI), Master APM Server, and Local APM Server
  - Master APM Server is only used on the master machine
- Follow basic guidelines
  - Plan once, build once